SHARP APL *for MVS*

Internal and Operational Changes

Version 21

SOLITON ASSOCIATES

SHARP APL is a registered trademark of Soliton Associates Limited.
SVP, NSVP, and Mailbox are trademarks of Soliton Associates Limited.
All other products referred to in this document are trademarks or registered trademarks of their respective companies.

©1994 by Soliton Associates Limited — Printed in Canada
44 Victoria Street, Suite 2100, Toronto, Ontario, Canada M5C 1Y2

All rights reserved. No part of this work may be reproduced in any form or by any means without prior written permission of the copyright owner.

Soliton Associates Limited shall not be liable for errors contained herein or for incidental consequential damages in connection with the use of this material. The information contained within this document is subject to change without notice.

Publication code: MM-000-1201

Contents

Preface

Introduction v
Who Should Read This Guide? v
Conventions vi
Related Publications vii
Contacting Soliton Associates vii

1. Introduction

Overview 1–1
SHARP APL Products 1–1
Major Changes for Version 21 1–3
DISTNEWS Items 1–4

2. SHARP APL Changes

Overview 2–1
SHARP APL for MVS Quartermaster Facility (QMSF) 2–2
SVP Installation Verification Program 2–2
APL Configuration Parameters 2–2
ESCON Channel Support 2–3
Local Lock Logic 2–3
Active Workspace Names 2–4
Swap Scheduler Scans 2–4
Intrinsic Function Facility 2–4
Memory Allocation 2–4

3. File System Changes

Overview 3–1
New File Condense System Verb 3–2
New *condense* System Administrator Verb 3–2
New □*copy* Verb 3–2
Online Full Dump 3–2
Non-Interlocked Online Dump 3–3
Track Buffering for the File System 3–3
System Directory Files 1 *permit* and 1 *perfile* 3–4
File System Configuration Parameters 3–4
Ehancements to □*append* and □*lib* 3–5

4. Changes to SHARP APL Utilities

Overview 4–1
SHARP APL Debugging Facility (DEBG) 4–1
Scheduling Full Restores 4–1
Online Dumps 4–1
Workspace Utility (WSUT) 4–2
File System Online Utility (FUTL) 4–3
File System Backup Restore Utility (BSUT) 4–3
Generalized Auxiliary Processor Shell (GAPS) 4–4
SHARP APL High Speed Print Processor (HSPR) 4–4

5. Changes to APL-Coded Software

Overview 5–1
B-Task Scheduler 5–1
Daily Billing 5–1
Sort Processor (MSRT) 5–1
SHARP APL Interface to MVS JES (IMVS) 5–2
314159 *log* System N-task 5–2
SHARP APL Operator Environment 5–2
Account Reporting and Maintenance 5–3
SHARP APL Program Editor (APE) 5–3
Dump Analysis 5–5

Preface

Introduction

SHARP APL Version 21 for MVS ("V21") provides a number of enhancements to the SHARP APL product, resulting in improved performance, increased availability, and enhanced operability of the SHARP APL system. These changes include modifications and additions to the base APL system and to the file system, as well as changes to the SHARP APL utilities and other related software products.

The *Internal and Operational Changes* guide describes the changes to SHARP APL since Version 20 as they relate to the installation, configuration, operation, and maintenance of the SHARP APL system. It includes internal changes in the way SHARP APL is implemented which are transparent to users, but which might be apparent in the performance perceived by those users or by other systems that share the computing facilities.

This document, when used with the *Guide for APL Programmers*, completes a summary of all changes to SHARP APL and related products. Detailed procedures for upgrading to Version 21 are provided in the *Upgrade Guide*.

Note: Rather than reproduce existing material, references to related SHARP APL publications are supplied where applicable.

Who Should Read This Guide?

This guide should be read by personnel responsible for the operation and maintenance of SHARP APL, by programmers and analysts responsible for applications using SHARP APL, and by other interested SHARP APL users.

Conventions

The following conventions are used throughout this guide.

`⎕io←0`	All examples in this guide assume that the index origin noun is set to `0`.
abcde... *ABCDE....*	Examples of input to and output from SHARP APL are shown in the APL character set, `⎕av`, in which the first and second APL alphabets appear in lowercase and uppercase, respectively. For more information, see "The APL Character Set," below.
`constant width`	Examples of non-APL input and output (e.g., JCL) are represented in a `constant width` typeface.
passno	Argument variables appear in *italic* type.
[ ]	Optional arguments are shown between square brackets. For example, in the following APL statement, the argument *passno* is optional:

'filename' `⎕stie` *tieno [passno]*

The APL Character Set

The names of APL files and workspaces are constructed from the characters *a–z*, *A–Z*, 0–9, and the glyphs Δ and ∆. The characters *a–z* are referred to as the first alphabet, and appear as lowercase letters in this guide. The characters *A–Z* are referred to as the second alphabet, and are represented as uppercase letters.

Due to the wide variety of input and output devices available for SHARP APL, these alphabets may display differently on your terminal. For example, the first alphabet is often displayed as uppercase, while the second alphabet is displayed as underscored uppercase. To determine what the first and second alphabets look like on your terminal, display the locations `⎕av[86+⍳26]` and `⎕av[113+⍳26]`.

APL workspace and file utility selection statements represent these characters as follows:

In APL	In EBCDIC
first alphabet	A–Z
second alphabet	a–z
0–9	0–9
Δ	- (minus)
∆	+ (plus)

The display of names is controlled by the `EXEC PARM` parameter `LOWERCASE=YES|NO`. If `LOWERCASE=YES` is specified, then the above convention is also used to write to `SYSPRINT`. If `LOWERCASE=NO` is specified, then second alphabet APL characters are displayed in uppercase, and a second line is printed, without carriage advance, underscoring these letters.

Related Publications

The following manuals are distributed with SHARP APL Version 21. The system administrator should ensure that these manuals are available.

- *IMVS 1.0 Configuration, Operation, and Maintenance Guide*, publication code MM-014-0301
- *Internal and Operational Changes*, publication code MM-000-1201
- *Major DSECTs and EQUATEs*, publication code MM-000-1401
- *Messages and Codes*, publication code MM-000-1601
- *Guide for APL Programmers* publication code MM-000-1501
- *QMSF 1.0 Configuration, Operation, and Maintenance Guide*, publication code MM-013-0301
- *System Administrator's Guide*, publication code MM-000-1001
- *System Maintenance Guide*, publication code MM-000-1101
- *Upgrade Guide*, publication code MM-000-0201
- *Utilities Manual*, publication code MM-000-1301

Contacting Soliton Associates

Problems or questions regarding your SHARP APL system or SHARP APL applications software should be directed to the Soliton Associates Technical Services group, which can be reached using either of the following methods:

- SHARP APL Mailbox to `zdist`
- Internet mail to *zdist@soliton.com*

Comments concerning this guide or requests for additional technical documentation should also be directed to this group. Further details about system support can be found in chapter 14 of the *System Administrator's Guide*.

1 Introduction

Overview

SHARP APL for MVS Version 21 (V21) is a major upgrade to SHARP APL for MVS Version 20. Changes to SHARP APL which make up V21 include modifications and additions to the base APL system and to the file system, as well as some changes to the SHARP APL utilities and other related software products. These changes to the SHARP APL language, the SHARP APL system, and the SHARP APL environment will help your site provide efficient and reliable service to your user community.

SHARP APL Products

The following products are distributed with the V21 base system.

BSCR, file system crash recovery utility
BSLT, file selective tape copy utility
BSON, online file system
BSTC, file system tape copy utility
BSUT, file system backup restore utility
FUTL, file system online utility
HSPR, high-speed print processor
IMVS, SHARP APL interface to MVS JES
MDXF, DASD formatting utility
MSRT, batch sort utility
QMSF, quartermaster facility
SHOV, file conversion utility
SSVP, shared variable processor system
WSTC, workspace tape copy utility
WSUT, workspace utility

Since Version 20, some products have been distributed as individual upgrades or as completely new releases. The following products can be licensed separately as extensions to your base system.

ATH/F	ASCII Terminal Handler/Fast Connect. This new asynchronous terminal handler provides an interface between SHARP APL and an X.25 device or network.
SSQL 2.0	SHARP APL DB2 Auxiliary Processor. SSQL no longer requires a one-to-one relationship between the SSQL user's APL task and DB2 — the number of users is constrained only by the maximum number of shared noun users and the size of shared noun storage. This version of SSQL allows access to DB2 locally or remotely, depending on the connection between users and DB2.
IDSH 2.0	Information Display Station Handler. IDSH version 2.0 includes 31-bit addressing for memory use above the 16-Mb line, expanded trace facilities for improved diagnostics and problem-solving, improved character support, and some new or improved commands.
NSVP	Network Shared Variable Processor. NSVP is a multi-user auxiliary processor that enables program-to-program communication between connected systems using an extended version of SHARP APL's SVP, Shared Variable Processor, protocol.
Performance Package	The V21 Performance Package combines three new facilities (Workspace Cache, TCB/Slot, and the application table) that work together to provide reduced reaction time, reduced CPU consumption, increased resource control, and increased file system control: The workspace caching facility, **WSCACHE**, allows swapping of workspaces to a configured mainstore cache, thus reducing the number of swaps to DASD. As a result, APL reaction time is reduced, IPRDY and TORUN queues are shortened, and file operation and SVP operation rates are increased. Performance is further improved through the use of **TCB/Slot**. This restructuring of the MVS subtask hierarchy within SHARP APL allocates one interpreter task (or TCB) for each workspace slot, resulting in the ability to directly control the priority of the APL task resident in a given slot. Furthermore, this new task structure allows for a more efficient posting mechanism that reduces CPU overhead for APL. The **application table** provides improved resource control by allowing the system administrator to associate priorities with individual segments of an application. Rather than adjusting on a task basis, the system administrator can now declare effective priorities for application processing segments.

MVSLink	MVSLink provides an interface from a SHARP APL session running on MVS or UNIX to single user APs running independently in MVS, such as SSQL ,VSDI, TSIO, and FCAP.

Major Changes for Version 21

The following list provides a summary of the major changes and additions to SHARP APL that make up V21. For detailed information on the changes themselves, refer to the appropriate section in this manual (or in another SHARP APL manual, if specified).

New System Verbs: File Condense and `condense`

The *file condense* privileged verb reorganizes a fragmented SHARP APL file even if the file is currently in use. The `condense` system administrator verb is used to perform a file condense of a complete file system volume class.

Improved Console Message Processing

APL-coded SHARP APL for MVS components now send messages to the MVS console rather than the SHARP APL OPR1 console.

ESCON Channel Support

ECKD (Extended Count Key Data) command set is now supported, allowing SHARP APL for MVS to take advantage of ESCON channel technology.

Improved File System EXCP Logic

The V21 SHARP APL file system utility EXCP logic that performs track buffering has been modified to use the same scheme employed by the new track buffering logic of the on-line file system. This enables a full track to be read in one-half less revolution than the logic previously employed.

File System Track Buffering

The V21 SHARP APL file system contains an optional facility called *track buffering* which provides an extra level of buffering in addition to standard file system buffering.

Non-Interlocked On-Line Dumps

The V21 file system backup restore utility (BSUT) allows on-line full dumps and on-line incremental dumps to be optionally non-interlocked. The option is separately selectable for each file system library dumped.

On-Line Full Dump

The V21 file system backup restore utility (BSUT) supports on-line full dump of a SHARP APL file system volume class. On-line full dump allows dumps to be taken while the volume class being dumped is in active use. The on-line mode takes about the same time as the off-line mode, and offers significant operational benefits.

Reduced QMS Resource Consumption

Resource consumption of QMS, the SHARP APL for MVS Quartermaster Facility, is significantly reduced.

DISTNEWS Items

General maintenance and improvements to SHARP APL since Version 20 provide a system that is more reliable, more efficient, and easier to use. These changes include DISTNEWS items 661, 674, 675, 712, 718, 735, 743, 746, 747, 749, 753, 759, 761, 764, 766, 767, 775, 778, 784, 785, 789, 791, 793, 794, 796, 790, 797, 798, 802, 803, 804, 806, 807, 808, 811, 812, 813, 814, 816, 817, 819, 822, 823, 824, 825, 828, 829, 830, 831, 836, 837, 839, 844, 845, 850, and 852. For information on these and other items, refer to the documentation in the `707 distnews` workspace on the Soliton Lab system.

2

SHARP APL Changes

Overview

The changes made to the SHARP APL system for V21 are outlined below. Some are discussed in more detail in the sections that follow this overview.

- QMSF, the SHARP APL Quartermaster Facility, is re-engineered to be more efficient.
- There are some new and changed APL configuration parameters. The `SJAP=` configuration parameter is eliminated.
- SHARP APL now supports primary/secondary workspace directories on any extent or pair of extents (implemented via the new `DIREXTENTS=` parameter).
- The ATASK control block contains the name and load time of the active workspace of every APL task.
- SVP, the Shared Variable Processor, is extended to support more than 16 active subsystems.
- The ECKD (Extended Count Key Data) channel command set is now supported.
- Interpreter now uses less critical system resources.
- Initiator failures generated by SIVP are resolved with redesigned CPU accounting code.
- V21 includes a reduction in the adverse effects of contention for the address space local lock.
- The new SHARP APL Intrinsic Function Facility introduces APL symbol/name association with routines implemented in Assembly language.

SHARP APL for MVS Quartermaster Facility (QMSF)

QMSF has been improved to reduce CPU consumption, and to avoid the problems caused by limited workspace size.

Previously, this AP consisted of a single privileged APL N-task. The improved facility now consists of an MVS-started task and a number of privileged APL N-tasks. The MVS-started task handles the application-independent parts of the quartermaster service, including shared noun interaction with the QMSF users. The APL N-tasks each perform different application-specific processing; for example, one N-task handles HCPRINT requests, another N-task handles BTASK requests, and so on.

QMSF CPU consumption is reduced because much of the code has been rewritten in Assembly language and many expensive SVP operations are moved from APL to Assembly-coded tasks. Also, the many applications handled by QMSF are no longer competing with each other for scarce resources within a single APL N-task.

SVP Installation Verification Program

The Shared Variable Processor Installation Verification Program (SIVP) previously ended abnormally, executing the aptest verification. The failure mode varied, but most frequently there was a system error in the initiator. This failure was caused by a changed definition in the ESA CPU timer code. Reorganization of CPU accounting code in SHARP APL should now allow SIVP to complete successfully.

APL Configuration Parameters

The following APL configuration parameters are new, changed, or no longer required in the SHARP APL PARMLIB member. See the SHARP APL *System Maintenance Guide* for complete descriptions of the new and changed parameters.

New: `DIREXTENTS=`

This parameter indicates the extent or extents upon which the directories are allocated. Each value specified must correspond to a value specified to WSUT at the time of the last workspace library system restore. For more information, please refer to the documentation on the workspace utility in Chapter 4.

New: `NSVPTIME=`

This new parameter determines the timeout interval in seconds for a number of NSVP actions. The same interval is used for all actions.

Changed: FILESIZE=

Two storage areas are now reserved by the FILESIZE= parameter: the first value defines the file size "below the line" and the second value defines the file size "above the line." Both values must appear in this argument.

Changed: PRIORITY=

The APL priority mechanism is expanded to add flexibility and avoid priority resonances between multiple APLs running on the same machine. The new syntax of the PRIORITY= parameter must now include a percentage of time for each priority specified:

PRIORITY=((*pct1,pri1*) , . . . , (*pct8,pri8*))

Where *pct,pri* represents 1 to 8 percentage/priority pairs. All percentages must lie in the interval (1,100), and their sum must be exactly 100. All priorities must lie in the range (1,255).

Removed: SJAP=

This configuration parameter has been eliminated because the System Journal Auxiliary Processor (SJAP) is no longer part of the SHARP APL system. For alternatives to SJAP, please contact Soliton Technical Services.

ESCON Channel Support

SHARP APL now supports the ECKD (Extended Count Key Data) channel command set. The channel programs used for workshop libraries, swap, and the file system now take advantage of this new feature.

Local Lock Logic

APL uses the address space Local Lock to ensure that multiple TCBs do not destroy each other's work by lack of synchronization. When a TCB enters non-reentrant code, it usually obtains the lock. If the lock is not available, the TCB gets suspended and, later, redispatched.

The overhead associated with suspension and redispatch is normally high; this is reduced by waiting in a CPU spin loop for a limited amount of time hoping that the lock is freed by the time the spin is complete. If the lock is obtained before the end of the CPU spin, overhead is saved with little cost.

Active Workspace Names

Each APL task now carries with it in its ATASK control block the name of its current active workspace (or 0 `clearws` in the case of a clear workspace) and the time at which the workspace was loaded or cleared. A privileged APL task can readily extract this information from the ATASK table, allowing the system administrator to track workspace usage. See the SHARP APL *System Administrator's Guide* for more information.

Swap Scheduler Scans

The SHARP APL swap scheduler is restructured to consume fewer resources and react quicker to changes in the time sharing environment. The scheduler also preferentially uses swap data sets on faster DASD devices.

Intrinsic Function Facility

The Intrinsic Function facility, new with V21, permits association of Assembly language routines with APL workspace objects. For more information on intrinsic functions, please refer to the V21 *Guide for APL Programmers.*

Memory Allocation

Memory allocated `BELOW` is actually allocated `BELOW,ANY` thus allocating memory which is "virtually" below the 16 Mb line, but which may be backed anywhere. This characteristic is very important for relief of low-memory constraints.

3 File System Changes

Overview

This chapter lists changes involving the SHARP APL file system. The changes outlined below. Some are discussed in more detail in the sections that follow this overview.

- There are some new and changed file system configuration parameters.
- SHARP APL's requirement for real memory below the 16 megabyte line is reduced. (Refer to the new `TRKBUFS=` parameter.)
- RPTSUB is replaced by the MVS macro `TRKCALC`
- New system verbs for V21:
  - File Condense.
  - *`condense`*.
  - □*`copy`*.
- Track buffering is now selectable as an option for the file system.
- File full dump can now be run offline or online.
- Non-interlocked online dump is now an option.
- The system verbs □*`append`* and □*`lib`* have been enhanced.
- Organization of the system directory files `1` *`permit`* and `1` *`perfile`* is made more effcient.

New File Condense System Verb

The new file condense privileged system verb, `□ff 43` reorganizes a fragmented SHARP APL file, even if the file is currently in use. The operation of this verb is functionally transparent to users of the file upon which it operates, although degraded response time may be observed due to contention for the file.

This verb begins by copying its source file into a newly created sink file. On termination of the copy, the source file is erased, the users of the source file are transferred to the sink file, and the sink file is renamed to become a new version of the source file. The new version is stored contiguously, if possible, and storage overhead, introduced by replacing file components of different sizes, is eliminated. Please see Chapter 11 of the SHARP APL *System Administrator's Guide* for documentation of the `□ff`.

New `condense` System Administrator Verb

The `condense` system administrator verb is an APL verb used to perform a file condense of a complete file system volume class. In some environments this can eliminate the need to run the BSUT restore job for the volume class.

New `□copy` Verb

`□copy`, the *file copy* system verb, is used to create a copy of a SHARP APL file even when the source file is currently in use. `□copy` is fully documented in the V21 *Guide for APL Programmers*.

Online Full Dump

The file system backup restore utility (BSUT) can perform a full dump of a file system volume class which is online to users. Online full dump takes about the same time as offline full dump, and offers significant operational benefits.

Previously, it was necessary to execute the BSUT offline full dump and restore jobs for each file system volume class on a weekly or monthly basis. The restore job was required to reorganize fragmented file space and recover overhead due to replacement of file components of different size. The full dump job was required to create the tapes for the restore job, and it was critical that the restore tapes matched the volume class exactly as it was at shutdown.

Depending on the size of the file system volume class involved, the offline execution of the full dump and restore jobs may take a number of hours, during which time the volume class is unavailable to users. The benefit of online full dump is that it reduces the time the volume class is offline by approximately one half.

In an offline full dump, all files in all file libraries are synchronized and dumped at the same moment. In an online full dump, all files in each file library are synchronized and dumped at the same moment, but different libraries are not synchronized, and are dumped at different moments. However, a fully synchronized dump can be obtained by performing an online full dump followed by an offline incremental dump. Under V21, a SHARP APL for MVS installation may choose to use the `condense` system administrator verb instead, eliminating the BSUT restore job altogether.

See the SHARP APL *Utilities Manual* for more information on BSUT.

Non-Interlocked Online Dump

BSUT ensures the integrity of each file library in an online full dump or online incremental dump by preventing changes to the files in the library during the time the library is being dumped. In the default mode of operation, users may experience an interlock delay while a library is being dumped.

The `COPY=` option, similar to the verb `⎕copy`, allows you to select no interlock delay while a file library is being dumped; however, response time may be degraded due to contention. This option may be applied to all libraries or to selected libraries.

When an online dump uses the non-interlocked method, it executes more slowly than with the interlocked method and requires more temporary disk storage; but, it offers the benefit that a file library may be dumped without significantly imparing its availability.

See the SHARP APL *Utilities Manual* for more information on BSUT.

Track Buffering for the File System

Track buffering is now selectable as an option for the online file system (BSON); previously it was used only by the file system utility (BSUT). Track buffering provides an extra level of buffering in addition to standard file system buffering.

When track buffering is selected via the `TRKBUFS=` configuration parameter, complete disk tracks are buffered in an area called the track buffer. Each time the file system requires a file system record that is not already present in a file system buffer, the required record is fetched from the track buffer area. If the disk track containing the required record is not present in a track buffer, the complete disk track containing the required record is read into the track buffer area. The file system automatically monitors whether track buffering is beneficial, on an extent-by-extent basis, and disables track buffering for those extents where it proves non-beneficial.

See the SHARP APL *Utilities Manual* for more information on BSON and BSUT.

System Directory Files `1 permit` and `1 perfile`

The organization of system directory files has been changed as follows:

A re-organized `1 perfile` is built by `⎕replace` rather than `⎕append`. This allows space outside the preallocated area to be re-used, preventing loss of disk space during re-organization.

File `1 permit` components are put on a free chain by `⎕erase`, rather than simply being lost. This removes the need for re-organization of `1 permit`, and the resulting loss of records.

See the SHARP APL *System Administrator's Guide* for more information on these system files.

File System Configuration Parameters

The following file system configuration parameters are new or changed. See the SHARP APL *System Maintenance Guide* for complete descriptions.

New: `TRKBUFS=`

This new parameter allocates a set of full track buffers above the 16Mb line. See the section "Track Buffering for the File System" in this chapter for more information.

New: `PHRPT=`

This new parameter is used to determine the number of records (of length `MAXRECL`) in a full track buffer. See `TRKBUFS=` for a description of full track buffering.

New: `DEBG=`

This new parameter specifies the details of the debug system programmer interface, used primarily for problem diagnosis.

Changed: `VC=`

The `TYPE=` sub-parameter to `VC=` no longer accepts `READONLY` as an option choice.

Ehancements to $append and $lib

Round robin record allocation during $append is improved; allocation is always from a non-empty extent table, advancing the extent pointer past any empty extent tables.

$lib scheduling logic is improved to reduce frequency of suspensions with system files interlocked.

4

Changes to SHARP APL Utilities

Overview

This chapter lists general and specific changes involving SHARP APL utilities. For detailed information on the function and use of SHARP APL utilities, refer to the SHARP APL *Utilities Manual*.

SHARP APL Debugging Facility (DEBG)

DEBG interface support has been added to all utilities. Although DEBG has not changed, an effort has been made to standardise `DEBG=` parameter arguments for most SHARP APL products. This facility is generally not used by customers unless DEBG information is required by Soliton support staff.

Scheduling Full Restores

A new method of eliminating fragmentation and checkerboarding of file volume classes has been introduced. This can considerably reduce the need to perform regular restores. See the SHARP APL *System Administrator's Guide* for documentation on the `condense` function for more information.

Online Dumps

Online dumps (both full and incremental) require that the APL system be running, the volume class be started (i.e., online), and a BG N-task be available to co-ordinate taking the dump. The result of an online dump cannot be used directly as input to the file system backup restore utility (BSUT) for performing RESTORE, the file system online utility (FUTL) for performing RETRIEVE, or the file

system selective tape copy utility (BSLT). However the file system tape copy utility (BSTC) automatically converts online tapes to a form that is suitable for these utilities.

When performing online dumps, an attempt is made to preserve the integrity of libraries, either by suspending modifications to file components with that library or by using the copy feature (see BSUT option COPY=YES/NO). However other libraries can be modified while the target library is being dumped, so an instantaneous snapshot of the entire volume class is not obtained.

Thus if a copy of an online dump (or incremental dump) is used as primary input to RESTORE, a warning message is issued, and an operator reply required, before proceeding. However an online dump may be used as secondary input providing the primary input is an offline incremental dump, and the resulting volume class will be a precise image of the volume class at the time of the incremental dump.

Workspace Utility (WSUT)

Within the APL workspace library system, the APL directories hold account information (such as encoded passwords) and pointers to each of the workspaces in a given library. Because of the critical nature of this information, two copies are kept of each directory. Previous versions of SHARP APL kept these two versions on the same "extent," namely extent 0. See the new configuration parameter DIREXTENTS= for more information.

Changed COMMAND Statements: CREATE and SELDUMP

In addition to tapes written by DUMP, or INCDUMP and DUMP, the input to CREATE may be the output of SELDUMP, provided that option DIRECTORY=YES was specified for the SELDUMP command.

New Configuration Parameter: DIREXTENTS=

The default behaviour of V21 APL is to continue to store both copies of the directories on extent 0, but the DIREXTENTS= parameter now allows them to be placed both on some other extent, or on two different extents.

This parameter consists of one or two values, each an extent number. If DIREXTENTS= is omitted, the default of (0,0) is used. If only the first value specified, the second value defaults to the first. For RESTORE or CREATE, the values define the extent numbers of the extents upon which the directories are allocated (for recoverability reasons each directory is written twice). For DUMP, INCDUMP, and SELDUMP, the values define the extents which directories are read from and written to.

File System Online Utility (FUTL)

New OPTION: PERMIT=

If PERMIT= is specified when PERMCN= is specifed on an INCLUDE statement, files included on behalf of that card are given permission matrices taken from the file named by option PERMIT=, using the specified component.

This option is primarily intended as a security measure when transporting files between APL systems.

New INCLUDE: PERMCN=

This specifies the number of the permission component within the file named by PERMIT=, to be used with the file, or set of files dumped on behalf of this include statement.

File System Backup Restore Utility (BSUT)

New OPTION/INCLUDE: COPY=

There are now two utility control statements available for use: OPTION COPY=(YES/NO) and INCLUDE ID=*lib*, COPY=(YES/NO). The COPY= keyword on an OPTION command sets the default technique used for all libraries. If YES, the utility will copy entire libraries instead of using a library interlock. The COPY= keyword on an INCLUDE statement overrides this default for specific libraries.

New OPTION: DEDICATED=

To perform an online full dump or incremental dump specify option DEDICATED=NO. To perform an offline dump or incremental dump, specify option DEDICATED=YES. If the state (online or offline) of the volume class is not appropriate to the value selected for DEDICATED= then the utility will terminate with an error message.

Earlier versions of BSUT detected whether the volume class was online or offline, and acted accordingly.

Changed COMMAND: INCDUMP

INCDUMP uses the value of option DEDICATED= to determine which type of incremental dump is desired. If this does not match the state of the volume class, the job will terminate with an error message.

Changed COMMAND: NODUMP

NODUMP can be used with the volume class online or offline *provided* option DEDICATED= is specified appropriately.

Generalized Auxiliary Processor Shell (GAPS)

Several SHARP APL products are now built using a common infrastructure known as Generalized Auxiliary Processor Shell (GAPS). GAPS implements some familiar facilities (e.g., the transfer vector) in a fashion that invalidates many of the familiar DUMP environment facilities and replaces them with more powerful new ones. Some of the distributed SHARP APL products created using GAPS structure include the Quartermaster facility (QMSF), the SHARP APL interface to DB2 (SSQL), and MVSLink.

SHARP APL High Speed Print Processor (HSPR)

HSPRINT no longer "auto-advances" to a new page after 250 lines of output when form depth is set to "infinite." The latest version of HSPR includes the addition of ⋸ (epsilon underbar) to HSPRINT translate tables.

5 Changes to APL-Coded Software

Overview

This chapter discusses changes involving APL-coded software such as the B-task scheduler, the sort processor, daily billing, the SHARP APL interface to MVS JES (IMVS), the `314159 log` system N-task, the operator environment, resource administration facilities, the SHARP APL Program Editor (APE), and the dump analysis system.

B-Task Scheduler

The algorithmn for calculation of flags is changed to operate with Boolean values. This modification reduces space requirements by a factor of 8 and cpu cost by a factor of 2.

Daily Billing

Sites can now specify an offset hour so billing runs can be started at any time (not just midnight).

Sort Processor (MSRT)

The `sortreq` verb now supports the left argument to character upgrade as specified via `maskfile=`*filename* `maskcomp=`*n*. A modification to MSRT now allows "active" sort requests to be withdrawn without an abnormal termination.

SHARP APL Interface to MVS JES (IMVS)

Several enhancements and fixes are provided with the V21 release of IMVS:

- IMVS is now more robust due to improved handling of unexpected JES events:
- There is now provision for processing a `/*JOBFROM` card for sites which require this for an access control facility.
- *mpp* classify logic is restuctured to allow for specifying Union style translation from EBCDIC by beginning a form name with the letter 'U'.
- A new facility in the *mrd* interface allows the submittor of a SYSIN file to provide one or more translate tables to be used in translating the file as it is passed along to JES. This is done by putting a component of the form `1⊃⎕av[mapping]` into the file. The translation takes effect as of the next text component.
- A card-scan exit is added to permit sites to validate or edit jobs submitted via *mrd*.
- Cloneid for *mrd* is changed from 8881 to 881.
- System tasks *mpp* and *mrd* now log messsages to the MVS console instead of the SHARP APL OPR1 console.

314159 *log* System N-task

The LOG N-task now automatically increases the file reservation limit of files which require resizing. This change avoids infinite wait for manual intervention causing file reservation to be exceeded.

SHARP APL Operator Environment

The following list is a summary of changes to the operator environment for V21. For more detailed information on the operator environment, please see the SHARP APL Administrator's Guide.

- The *perflib* verb is modified to be significantly faster.
- The APL priority mechanism is expanded to add flexibility and avoid priority resonances between multiple APLs running on the same machine. The *priority* verb an empty argument or a closed array argument of up to eight percentage/priority pairs:
- The *aschap* verb now accepts one of two non-empty arguments: 'on' and 'off'. It's sole verb is to enable or disable the APL address space CHAP.
- The new *damax* verb is added for setting the maximum active workspace size permitted to be swapped to a given extent.
- The new *reschhz* verb is added for setting and reporting the frequency at which the APL scheduler is automatically posted.

- The new *desuspend* verb is added for desuspending APL tasks whose execution was suspended using the *suspend* verb.
- The *segpri* verb is added to 314158 *opfns* for setting the application priority for application segments.
- The *Setseg* verb is added to 314158 *opfns*, to be used by programmers for accessing the application table to enter application segments.
- The *suspend* verb is added to 314158 *opfns* for suspending execution of APL tasks.
- The *taskpri* verb is added to 314158 *opfns* for setting the intrinsic priority of APL tasks.
- The *publicwss* verb is added to 314158 *opr* for returning the names of the public workspaces owned by the specified account number(s).
- The *sob* verb is added to 314158 *opr* to display information on system performance behavior.

Account Reporting and Maintenance

The following changes to account reporting and maintenance facilities are provided with V21:

- The CPU limit displayed by the *quotas* verb was 1/10th the actual limit; this display is now accurate.
- The size at which the initial *numbreq* file is generated limits the number of directories to 30, which means that the system is limited to 9000 requests. The file generation mechanism is changed to increase the limit of requests to 150,000.

  This generation is available for new sites only. If current sites wish to increase the number of directories in their *numbreq* file, they should contact Soliton's Technical Services group for assistance.

SHARP APL Program Editor (APE)

A few enhancements and minor bug fixes are provided with the V21 release of APE, including DISTNEWS item 718. See Chapter 4 of the *System Maintenance Guide* for more detailed information on any of the APE commands listed below.

- This release of APE has fixed a rank error occuring in *fna* when *lsl* contains 0 or negative numbers upon execution of the ape command *sl*.
- Several changes to APE are included to help in rationalising character set usage:
  - there is general control over the fullscreen drivers to provide case folding under control of a *termtype* noun.

- APE commands are now case invariant (that is, the two-letter command codes may be typed in any case)
- APE fullscreen directives (i.e., *A, B, I, D*) may now be typed in any case.
- assembly jobs now come back in Union font.
- case dependancies are removed from the *paper* workspace (e.g., searches in *vlist* are now case-insensitive).

- C programs can now use free format rather than the convention of 80 column images.

APE Commands

The following APE commands are given alternate names for V21:

- *pz* is now also called *pa*, meaning **pa**use.
- *lu* is now also called *st*, meaning **st**op, or *qu*, meaning **qu**it.

New Command: *fc*

The *fc* (**f**old **c**ase) command, for migrating documents between character standards, converts the case of characters based on information specified in its arguments.

Changed Command: *te*

The *te* command has a new option, *caps*, that indicates or changes case setting.

New Command: *ft*

The *ft* (**f**ormat **t**ext) command reformats lines of text in the workfile.

Changed Command: *sa*

This command now has different searching methodology. It can be given the component number of an audit locator to display the whole thing, or given a DSE and to only display hits on the DSE (regardless of what else might be in the audit locator).

Removed Command: *eu*, editor utility

This command is eliminated.

Changed Command: *qy*, prepare and queue MVS job

A job document can now be used as the first name in the doc list. If provided, this job document is used in place of the default job document for all jobs prepared in that execution of *qy*. Documents which fail job preparation or submission now cause an error message to be output to the submitter.

Dump Analysis

A new application group for products built using the Generalized Auxiliary Processor Shell (GAPS) is included with the V21 dump analysis system. GAPS implements some familiar facilities (e.g., the transfer vector) in a fashion that invalidates many of the familiar dump environment facilities and replaces them with new more powerful ones.

See the "System Maintenance Guide" for descriptions of the new verbs.